

# PYTHON İLE PROGRAMLAMANIN TEMELERİ (1.DERS NOTLARI)

## Konu Başlıkları

- Neden Python Programlama Dili
- print() Fonksiyonu
- Değişken Kavramı
- Temel Veri Türleri
- print ( ) Fonksiyonu ile Parametre Kullanımı
- Format Metodu ve
- f-string Kullanımı

### Python Kurulumu Nasıl Yapılır?

<https://www.youtube.com/watch?v=Na770QFsB5E>



## 1. Neden Python Programlama Dili

**Python**, öğrenmesi kolay, tamamen özgür ve ücretsiz bir programlama dilidir. Nesne yönelimli bir dil olup okunabilirliği yüksektir. Python dili başka programlama dilleri ile kıyaslandığında işlemleri daha az kod ile yapmak mümkündür. Python, bütün işletim sistemleri ile uyum içerisinde çalışmaktadır. (linux, macos, pardus) Python ile bilgisayarlar, mobil cihazlar ve web için uygulamalar geliştirmek mümkündür. Özellikle yapay zekâ, makine öğrenmesi ve derin öğrenme alanlarında Python dili oldukça yaygın olarak kullanılır.

## 2. print() fonksiyonunun kullanılması

**print()** fonksiyonu, istenilen bilginin ekrana yazdırılması için kullanılır. Bu bilgi bir karakter dizisi (metin), bir sayı ya da bir değişken olabilir.

```
print("Merhaba")
```

Merhaba

```
print(75)
```

75

```
sehir = "Kocaeli"  
print(sehir)
```

Kocaeli

**print()** fonksiyonunda yazdırılacak veri sayısı birden fazla ise aralarına virgül işareti konulmalıdır.

### KULLANIMI

**print(veri1, veri2, veri3, .....)**

```
ad = "Ahmet"  
soyad = "ÇELİK"  
yas = 14  
  
print(ad, soyad, yas)
```

Ahmet ÇELİK 14

Metin veya yazılar "....." Çift tırnaklar arasına yazılabileceği gibi '.....' tek tırnaklar arasına da yazılabilir.

```
print('15 Temmuz Şehitler Fen Lisesi')
```

15 Temmuz Şehitler Fen Lisesi

**NOT:** Sayılar tırnak içinde yazılmaz. Sayılar tırnak içerisinde yazılırsa metin olarak algılanır ve bu verilerle matematiksel işlem yapılamaz.

**print(""" """)** Üç tırnak Kullanımı

Print fonksiyonunda üç tırnaklar arasında yazılan içerik, aynı düzen ve biçimde çıktı olarak ekrandan alınacaktır.

```
print("""
-----
|           Sevgili Öğrenciler           |
|           OKULUMUZA HOŞGELDİNİZ       |
|-----|
""")
```

```
-----
|           Sevgili Öğrenciler           |
|           OKULUMUZA HOŞGELDİNİZ       |
|-----|
```

Python'da Kodların yanına açıklama ve yorum yazmak için # sembolü kullanılır. Yorumlar kod olarak değerlendirilmezler ve çıktı olarak gösterilmezler.

```
print("Bu bir çıktıdır.")
# Bu bir açıklamadır.
```

Bu bir çıktıdır.

### 3. Değişken Kavramı

Değişkenler (veri tutucular) kendisine değer atayabileceğimiz ve verileri saklayabileceğimiz nesnelerdir.

**ad**, **soyad**, **yas** ve **sinif** birer değişkendir. Değişkenlere değer ataması "=" işareti ile yapılır.

```
ad = "Serra"
soyad = "ÖZTÜRK"
sinif = "9-F"
yas = 15

print("Öğrencinin Adı   :",ad)
print("Öğrencinin Soyadı:",soyad)
print("Öğrencinin Sınıfı:",sinif)
print("Öğrencinin Yaşı  :",yas)
```

Öğrencinin Adı : Serra  
Öğrencinin Soyadı: ÖZTÜRK  
Öğrencinin Sınıfı: 9-F  
Öğrencinin Yaşı : 15

```
ad = "Ahmet"
soyad = "YILMAZ"
print(ad, soyad)
```

Ahmet YILMAZ

```
sayi1 = 45
sayi2 = 34
sonuc = sayi1 + sayi2
print("Toplam = ", sonuc)
```

Toplam = 79

## 4. Temel Veri Türleri

Python'da kullanılan çok sayıda veri türü bulunmaktadır. Bu veri türleri ilerleyen konularda ele alınacaktır ancak temel olarak üç veri tipi kullanılmaktadır. Bunlar metin, tam sayı ve ondalıklı sayı veri tipleridir.

```
il = "Kocaeli"
ilce = "Çayirova"
mahalle = "Akse"
telefon = "0262 721 0715"
```

### Karakter Dizileri (Metinsel Veriler - String) – str

Tırnak içerisinde yazılan her şey String veri tipindedir. Karakter dizisi, yani metin-yazı olarak algılanır. Kısaca **str** olarak belirtilir.

```
yas = 16
ogrenciMevcutdu = 486
not1 = 95
taksitSayisi = 36
gunSayisi = 22
```

### Tam Sayılar (Integer) – int

Tam sayılardır. Pozitif, negatif ya da sıfır değeri alabilir. Noktalı sayı değeri içermez. Kısaca **int** olarak belirtilir

```
devamsizlik = 4.5
ortalama = 97.75
fiyat = 49.90
kdvOran = 0.18
```

### Ondalıklı (Noktalı) Sayılar – float

Noktalı sayılar da denir. **float** olarak belirtilir. Ancak burada virgül yerine nokta kullanmamız gerekir.

## 5. print ( ) Fonksiyonu ile Parametre Kullanımı

### \n (Alt Satır) Parametresi

Print fonksiyonu kullanılırken alt satıra inme ihtiyacı duyulduğunda "\n" parametresi kullanılabilir.

```
print("1.Satır\n2.Satır\n3.Satır")
```

1.Satır  
2.Satır  
3.Satır

```
print("Adı: Harun\nSoyadı: ÖZTÜRK")
```

Adı: Harun  
Soyadı: ÖZTÜRK

### \t (Tab – Sekme) Parametresi

Klavyedeki Tab tuşuna basıldığında gibi belli karakter boşluk bırakmak için kullanılır.



```
print("Adı\t: Harun")
print("Soyadı\t: ÖZTÜRK")
```

Adı : Harun  
Soyadı : ÖZTÜRK

## sep Parametresi

print fonksiyonu içerisinde aralarında virgül (,) olan birden fazla değer yer alabilir. sep parametresi bu değerler ekrana yazdırılırken değerlerin arasına belirlediğimiz karakterin yazılmasını sağlar. sep parametresinin varsayılan değeri boşluk (" ") karakteridir.

Aşağıda sep parametresi kullanılmadan 3,4 ve 5 değerleri ekrana yazdırılmıştır. Çıktıda aralarında varsayılan değer olan boşluk yer almaktadır.

```
print(3,4,5)
```

3 4 5

Şimdi sep parametresini kullanarak değerler arasındaki karakteri "-" (tire işareti) belirleyelim.

```
print(3,4,5,sep="-")
```

3-4-5

```
gun = 11  
ay = 1  
yil = 2021  
print(gun,ay,yil,sep = "/")
```

11/1/2021

```
print("Harun", "ÖZTÜRK", 2006, sep="\n")
```

Harun  
ÖZTÜRK  
2006

## end Parametresi

print fonksiyonu ile ekrana yazdırılan değerlerin sonunda hangi karakterin yer alacağını belirler. end parametresinin varsayılan değeri "\n" (alt satıra geç) dir.

```
print("Çayırova")  
print("KOCAELİ")
```

Çayırova  
KOCAELİ

```
print("Çayırova",end=" ")  
print("KOCAELİ")
```

Çayırova KOCAELİ

```
print("Çayırova",end="/")  
print("KOCAELİ")
```

Çayırova/KOCAELİ

## 6. Format Metodu

format metodu bir çıktının içerisine önceden tanımlı değerleri (değişkenleri) yerleştirmek için kullanılır.

**print ( )** fonksiyonu ile **.format** metodu ve **{ }** parametresi kullanımı :

```
print("{} yazılar {} yazılar {} yazılar".format(degisken1,degisken2,degisken3))
```

**{ }** – süslü parantezlerin bulundukları yerlere sırasıyla *degiken1*, *degisken2* ve *degisken3* değerleri yazılır.

Önce format metodunu kullanmadan bir örnek yapalım.

```
sayi1 = 13
sayi2 = 17
toplam = sayi1 + sayi2
print(sayi1,"ve",sayi2,"sayilarinin toplamı:",toplam)
```

13 ve 17 sayilarinin toplamı: 30

Şimdi de aynı örneği format metodu kullanarak uygulayalım.

```
sayi1 = 13
sayi2 = 17
toplam = sayi1 + sayi2
print("{} ve {} sayılarının toplamı: {}".format(sayi1, sayi2, toplam))
```

13 ve 17 sayılarının toplamı: 30

```
isim = "Melda ORHAN"
sinif = "9/A"
no = 803
print("""Okulumuz {} sınıfı {} numaralı öğrencisi
{}
bu belgeyi almaya hak kazanmıştır.""").format(sinif,no,isim))
```

Okulumuz 9/A sınıfı 803 numaralı öğrencisi  
Melda ORHAN  
bu belgeyi almaya hak kazanmıştır.

**Değişkenlerin yazdırma sırasını belirleme**

```
a = 5
b = 10
c = 15
print ("{2},{0},{1}".format(a,b,c))
```

15,5,10

**Ondalıklı sayıların basamak sayısını belirleme**

```
a = 1.5485
b = 2.8765
c = 3.9321
print("{:.2f},{:.3f},{:.1f}".format(a,b,c))
```

1.55,2.877,3.9

## Hizalayıarak yazma

```
okul = "15 Temmuz ŞFL"
print("{:50}".format(okul))
print("{:^50}".format(okul))
print("{:>50}".format(okul))
```

15 Temmuz ŞFL

15 Temmuz ŞFL

15 Temmuz ŞFL

- : → hizalama yapılacağını belirtir varsayılan olarak yazı sola hizalı yazılır.
- 50 → kaç karakter ayrılacağını belirtir
- ^ → ayrılan karakter sayısı alanına yazıyı ortalı olarak yazar
- > → ayrılan karakter sayısı alanına yazıyı sağa hizalı olarak yazar

**NOT :** print ( ) fonksiyonu ile kullanılabilecek daha detaylı format bilgisine <http://pyformat.info> adresinden ulaşılabilir.



Using % and .format() for great good!

## 7. f-string kullanımı

Python 3.6 sürümü ile birlikte (şu an güncel sürüm 3.10) eklenen bir özelliktir. format metoduna alternatif olarak okunabilirliği arttırmak amacıyla geliştirilmiştir. format metodunda olduğu gibi bir çıktının içerisine önceden tanımlı değerleri yerleştirmek için kullanılır.

### Örnek:

```
sayi1 = 5
sayi2 = 7
toplam = sayi1+sayi2
print(f"{sayi1} ve {sayi2} sayılarının toplamı: {toplam}")
```

5 ve 7 sayılarının toplamı: 12

```
isim = "Melda ORHAN"
sinif = "9/A"
no = 803
print(f"""Okulumuz {sinif} sınıfı {no} numaralı öğrencisi
{isim}
bu belgeyi almaya hak kazanmıştır.""")
```

Okulumuz 9/A sınıfı 803 numaralı öğrencisi  
Melda ORHAN  
bu belgeyi almaya hak kazanmıştır.